

## **CLAIMS:**

1. (Currently amended) ~~Apparatus~~ An apparatus, having a processor, for workload balancing in an asynchronous messaging system comprising:

means for obtaining [[the]] an average queue depth of a message queue of ~~messages~~; [[and]]

means for controlling [[the]] a number of server instances of a server for retrieving messages from the message queue based on the average queue depth and one or more predetermined thresholds;

means for determining if the average queue depth exceeds a first predetermined threshold of the one or more predetermined thresholds;

means for initiating the start of a server instance for retrieving messages from the message queue by placing a trigger message on an initialisation queue in response to determining that the average queue depth exceeds the first predetermined threshold of the one or more predetermined thresholds, wherein the trigger message indicates that the server instance is to be started; and

means for resetting the average queue depth to less than the first predetermined threshold immediately upon the start of the server instance in response to determining that the first predetermined threshold has been exceeded.

2-6. (Cancelled)

7. (Currently amended) The apparatus of claim 1, wherein the means for controlling the number of server instances comprises:

means for terminating [[a]] the server instance when the average queue depth falls below a second predetermined threshold of the one or more predetermined thresholds.

8. (Currently amended) The apparatus of claim 7, comprising:

means, responsive to determining that the average queue depth is below the second predetermined threshold, for resetting the average queue depth to [[be]] greater

than the second predetermined threshold immediately upon the termination of the server instance.

9. (Currently amended) The apparatus of claim 7, wherein the means for terminating [[a]] the server instance comprises at least one of:

(i) means for ~~spoofing~~ indicating to the server instance ~~into believing~~ that there are no more messages to process on the message queue ~~for it to process~~;

(ii) means for ~~spoofing~~ indicating to the server instance ~~into believing~~ that a queue manager, controlling the message queue, is shutting down;

(iii) means for ~~spoofing~~ indicating to the server instance ~~into believing~~ that operator intervention ~~has requested~~ is requesting that the server instance [[shuts]] shut down;

[[and]] or

(iv) means for requesting that the server instance [[shuts]] shut down.

10. (Cancelled)

11. (Original) The apparatus of claim 1 comprising:

means for setting a maximum number of server instances that can be active at any one time.

12. (Original) The apparatus of claim 1, comprising:

means for setting a minimum number of server instances that should be active at any one time.

13. (Currently amended) The apparatus of claim 1 wherein the means for obtaining the average queue depth comprises:

means for calculating the ~~queue's~~ average queue depth of the message queue.

14. (Currently amended) The apparatus of claim 13 wherein the means for calculating comprises:

means for calculating a time weighted mean average queue depth of the message queue.

15. (Currently amended) The apparatus of claim 13, wherein the means for calculating comprises:

means for calculating an exponentially smoothed average queue depth of the message queue.

16-19. (Cancelled)

20. (Currently amended) A method for workload balancing in an asynchronous messaging system comprising:

obtaining ~~[[the]]~~ an average depth of a message queue ~~of messages~~; ~~[[and]]~~

controlling ~~[[the]]~~ a number of server instances of a server for retrieving messages from the message queue based on the average queue depth and one or more predetermined thresholds;

determining if the average queue depth exceeds a first predetermined threshold of the one or more predetermined thresholds;

responsive to determining that the average queue depth exceeds the first predetermined threshold, initiating the start of a server instance for retrieving messages from the message queue by placing a trigger message on an initialisation queue, wherein the trigger message indicates that the server instance is to be started; and

responsive to determining that the first predetermined threshold has been exceeded, resetting the average queue depth to less than the first predetermined threshold immediately upon the start of the server instance.

21-25. (Cancelled)

26. (Currently amended) The method of claim 20, wherein the controlling step comprises:

terminating ~~[[a]]~~ the server instance when the average queue depth falls below a second predetermined threshold of the one or more predetermined thresholds.

27. (Currently amended) The method of claim 26 comprising:  
responsive to determining that the average queue depth is below the second predetermined threshold, resetting the average queue depth to ~~[[be]]~~ greater than the second predetermined threshold immediately upon the termination of the server instance.

28. (Currently amended) The method of claim 26, wherein the step of terminating a server instance comprises at least one of:

- (i) ~~spoofing indicating to~~ the server instance ~~into believing~~ that there are no more messages to process on the message queue ~~for it to process~~;
- (ii) ~~spoofing indicating to~~ the server instance ~~into believing~~ that a queue manager, controlling the message queue, is shutting down;
- (iii) ~~spoofing indicating to~~ the server instance ~~into believing~~ that operator intervention ~~has requested~~ is requesting that the server instance ~~[[shuts]]~~ shut down; ~~[[and]]~~ or
- (iv) requesting that the server instance ~~[[shuts]]~~ shut down.

29. (Cancelled)

30. (Currently amended) The method of claim 20, further comprising:  
setting a maximum number of server instances that can be active at any one time.

31. (Currently amended) The method of claim 20, further comprising:  
setting a minimum number of server instances that should be active at any one time.

32. (Currently amended) The method of claim 20, wherein the step of obtaining the average queue depth comprises:  
calculating the ~~queue's~~ average queue depth of the message queue.

33. (Currently amended) The method of claim 32 wherein the step of calculating comprises:

calculating a time weighted mean average queue depth of the message queue.

34. (Currently amended) The method of claim 32, wherein the step of calculating comprises:

calculating an exponentially smoothed average queue depth of the message queue.

35. (Currently amended) A computer program for workload balancing in an asynchronous messaging system, the computer program ~~comprising~~ stored on a computer recordable medium having computer readable program code means adapted to perform the steps of, wherein the computer readable program code, when executed in a data processing system, causes the data processing system to:

~~obtaining the~~ obtain an average depth of a message queue ~~of messages~~; [[and]]

~~controlling the~~ control a number of server instances of a server for retrieving messages from the message queue based on the average queue depth and one or more predetermined thresholds;

determine if the average queue depth exceeds a first predetermined threshold of the one or more predetermined thresholds;

responsive to determining that the average queue depth exceeds the first predetermined threshold, initiate the start of a server instance for retrieving messages from the message queue by placing a trigger message on an initialisation queue, wherein the trigger message indicates that the server instance is to be started; and

responsive to determining that the first predetermined threshold has been exceeded, reset the average queue depth to less than the first predetermined threshold immediately upon the start of the server instance.

36. (New) The computer program of claim 35, wherein the computer readable program code to control the number of server instances further includes computer readable program code that causes the data processing system to:

terminate the server instance when the average queue depth falls below a second predetermined threshold of the one or more predetermined thresholds.

37. (New) The computer program of claim 36, wherein the computer readable program code further causes the data processing system to:

responsive to determining that the average queue depth is below the second predetermined threshold, reset the average queue depth to greater than the second predetermined threshold immediately upon the termination of the server instance.

38. (New) The computer program of claim 36, wherein the computer readable program code to terminate the server instance further includes computer readable program code that causes the data processing system to perform at least one of:

- (i) indicating to the server instance that there are no more messages to process on the message queue;
- (ii) indicating to the server instance that a queue manager, controlling the message queue, is shutting down;
- (iii) indicating to the server instance that operator intervention is requesting that the server instance shut down; or
- (iv) requesting that the server instance shut down.

39. (New) The computer program of claim 35, wherein the computer readable program code further causes the data processing system to perform at least one of:

setting a maximum number of server instances that can be active at any one time;

or

setting a minimum number of server instances that should be active at any one time.

40. (New) The computer program of claim 35, wherein the computer readable program code to obtain the average queue depth further includes computer readable program code that causes the data processing system to:

calculate the average queue depth of the message queue.

41. (New) A system, comprising:  
a processor; and  
a memory coupled to the processor, wherein the memory comprises instructions which, when executed by the processor, cause the processor to:  
obtain an average depth of a message queue;  
control a number of server instances of a server for retrieving messages from the message queue based on the average queue depth and one or more predetermined thresholds;  
determine if the average queue depth exceeds a first predetermined threshold of the one or more predetermined thresholds;  
responsive to determining that the average queue depth exceeds the first predetermined threshold, initiate the start of a server instance for retrieving messages from the message queue by placing a trigger message on an initialisation queue, wherein the trigger message indicates that the server instance is to be started; and  
responsive to determining that the first predetermined threshold has been exceeded, reset the average queue depth to less than the first predetermined threshold immediately upon the start of the server instance.
42. (New) The system of claim 41, wherein instructions for controlling the number of server instances further cause the processor to:  
terminate the server instance when the average queue depth falls below a second predetermined threshold of the one or more predetermined thresholds.
43. (New) The system of claim 42, wherein the instructions further cause the processor to:  
responsive to determining that the average queue depth is below the second predetermined threshold, reset the average queue depth to greater than the second predetermined threshold immediately upon the termination of the server instance.
44. (New) The system of claim 42, wherein the instructions for terminating the server instance further cause the processor to perform at least one of:

- (i) indicating to the server instance that there are no more messages to process on the message queue;
- (ii) indicating to the server instance that a queue manager, controlling the message queue, is shutting down;
- (iii) indicating to the server instance that operator intervention is requesting that the server instance shut down; or
- (iv) requesting that the server instance shut down.

45. (New) The system of claim 41, wherein the instructions further cause the processor to:

set a maximum number of server instances that can be active at any one time.

46. (New) The system of claim 41, wherein the instructions further cause the processor to:

set a minimum number of server instances that should be active at any one time.

47. (New) The system of claim 41, wherein the instructions for obtaining the average queue depth further cause the processor to:

calculate the average queue depth of the message queue.